

Article

A Two-Way Parallel Slime Mold Algorithm by Flow and Distance for the Travelling Salesman Problem

Meijiao Liu ^{1,2}, Yanhui Li ¹, Qi Huo ¹, Ang Li ^{1,2}, Mingchao Zhu ^{1,*}, Nan Qu ^{3,*}, Liheng Chen ¹ and Mingyi Xia ¹

¹ Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; liumeijiao17@mails.ucas.ac.cn (M.L.); liyanhui@ciomp.ac.cn (Y.L.); huoqi@ciomp.ac.cn (Q.H.); liang@ciomp.ac.cn (A.L.); chenliheng@ciomp.ac.cn (L.C.); xiamingyi@ciomp.ac.cn (M.X.)

² School of Optoelectronics, University of Chinese Academy of Sciences, Beijing 100049, China

³ College of Resources and Environmental Sciences, Jilin Agricultural University, Changchun 130118, China

* Correspondence: zhumingchao@ciomp.ac.cn or mingchaozhu@gmail.com (M.Z.); qunan@jlau.edu.cn (N.Q.)

Received: 9 July 2020; Accepted: 3 September 2020; Published: 5 September 2020



Featured Application: The design of this article can be applied to Travelling Salesman Problem and related problems.

Abstract: In order to solve the problem of poor local optimization of the Slime Mold Algorithm (SMA) in the Travelling Salesman Problem (TSP), a Two-way Parallel Slime Mold Algorithm by Flow and Distance (TPSMA) is proposed in this paper. Firstly, the flow between each path point is calculated by the “critical pipeline and critical culture” model of SMA; then, according to the two indexes of flow and distance, the set of path points to be selected is obtained; finally, the optimization principle with a flow index is improved with two indexes of flow and distance and added random strategy. Hence, a two-way parallel optimization method is realized and the local optimal problem is solved effectively. Through the simulation of Traveling Salesman Problem Library (TSPLIB) on ulysses16, city31, eil51, gr96, and bier127, the results of TPSMA were improved by 24.56, 36.10, 41.88, 49.83, and 52.93%, respectively, compared to SMA. Furthermore, the number of path points is more and the optimization ability of TPSMA is better. At the same time, TPSMA is closer to the current optimal result than other algorithms by multiple sets of tests, and its time complexity is obviously better than others. Therefore, the superiority of TPSMA is adequately proven.

Keywords: Slime Mold Algorithm; two-way parallel optimization; flow and distance; Travelling Salesman Problem

1. Introduction

The Travelling Salesman Problem (TSP) [1,2] is a classical problem in Non-Deterministic Polynomial problems, and has important practical significance in road network planning, workshop dispatch, and so on. With the development of the heuristic bionic algorithms and their good effect on solving problems, TSP has been solved by various intelligent algorithms such as Genetic Algorithm (GA) [3,4], Particle Swarm Optimization (PSO) [5,6], Ant Colony Optimization (ACO) [7,8], the classic heuristic Lin-Kernighan [9,10] and Lin-Kernighan-Helsgaun Solver (LKH) [11,12], etc. PSO is simple, but its effect is not good. The effect of GA is general and the algorithm is complex. ACO has a good effect, but its convergence speed is slow. For Lin-Kernighan, its complexity increases exponentially with the increase in the number of path points, so it takes too long to obtain results. LKH is an improved algorithm on the basis of Lin-Kernighan. Although the optimization method is the best so far in TSP, the time complexity is still too large to optimize quickly.

The microorganism called slime mold [13] has an evolutionary history of 100 million years. It can eventually find a high-quality foraging path by avoiding external obstacles in a network structure. The unique and efficient foraging process provides a new inspiration for path optimization. Therefore, scientists learned from the intelligent life and obtained a new algorithm called Slime Mold Algorithm (SMA) [14]. With the unique searching ability, slime molds can efficiently find a high-quality way to the food in maze experiments. As shown in Figure 1, based on the slime mold's foraging behavior, the scientists thought of the food sources as the cities of a country. It was found that the foraging path network of slime molds is highly similar to the actual road network designed in the country [15–18]. Thus, foraging behavior of slime mold has great significance to be explored for path optimization [19–23].

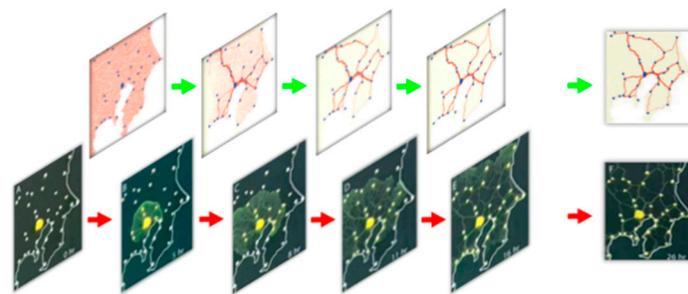


Figure 1. Foraging network of slime mold and road network.

In 2000, Japanese scientist Nakagaki [14,24–26] discovered that the slime mold can walk through the maze. The researchers placed the slime molds in a maze and dropped food at the entrance and exit of the maze in a petri dish. After a period of time, slime molds formed a feeding route. It was obtained as the solution to solve a complex maze problem. In 2007, Tero [27,28] came up with the model of “pipeline culture,” which mainly used Poisson Theorem and Kirchhoff’s Laws to realize the pipeline mechanism of flow and conductivity. This model can be equivalent to the foraging behavior of slime molds. Afterwards, Gunji et al. [29,30] applied the model to the description of networks in a cellular model. In China, Southwest University has studied the Slime Mold Algorithm combined with the pheromones of the Ant Colony Optimization to solve the classic TSP and multi-object TSP [31–34]. Compared with other algorithms, SMA has high-efficiency optimization ability, especially in solving the complex path problem, which includes a large number of points and complex distribution [35–37]. The way of SMA has made a new method to solve TSP. At the same time, the convergence of SMA is fast, due to fewer iterations. However, SMA has been researched more recently than the others, and there is still a lot of room to investigate and improve [1,2,27,28,33,34]. In summary, the bottlenecks besides its advantages in TSP are as follows:

- Due to the high similarity of some flow values, SMA cannot make a suitable choice. If points are selected only by flow, the ability will have a great limitation of global optimization.
- SMA has no randomness and the selected points and the points to be selected have strong correlations. Therefore, SMA has low flexibility and weak robustness.

The Two-way Parallel Slime Mold Algorithm (TPSMA) by flow and distance for TSP is proposed in this paper. TPSMA involves two indicators of flow and distance for path selection and adds random factors. TPSMA will improve the quality of SMA for solving TSP and achieve the following advantages:

- The selection rule combines two indicators of flow and distance, which makes SMA not only rely on the flow. It can better improve global optimization ability and prevent the algorithm from falling into local optimum.
- The proposed TPSMA adds random factors to increase the diversity of path choices and improve the robustness of the algorithm.

The structure of the rest of the article is: the second part describes the basic principle of SMA; the third part describes the design of TPSMA, including a specific, improved strategy; the fourth part, simulation and analysis; the fifth part, conclusions.

2. Slime Mold Algorithm

According to the foraging behavior of slime mold, the basic idea of SMA is: the slime molds expand to every direction and stretch themselves to cover the surroundings. Then, the slime mold will shrink back to the direction without food or far away from food, and they will continue to expand in the direction of food. That is, if they feel that the food is abundant, they will continue to expand; if they feel that the food is scarce, they will shrink and return. After a period of time, the slime molds will form a path, like a pipeline, and find the shortest route.

The “pipeline culture” model can be abstracted by slime mold foraging. The basic principle of SMA is [20–22,31–34]: firstly, pipeline paths are built in all directions and form a network between the food sources by imitating foraging behavior of slime mold; then, based on the length of each path, the width of pipeline, and the obstacles on path, foraging paths can be obtained; thirdly, under the effect of iteration, the stable distribution of flow will be formed through a period of dynamic transformation; finally, a path is generated from the start position to the end that has the food. The schematic diagram of SMA is shown in Figure 2. According to the slime mold algorithm in solving TSP, we obtain the “pipeline culture” model [2,31–37], and the specific algorithm is as follows:

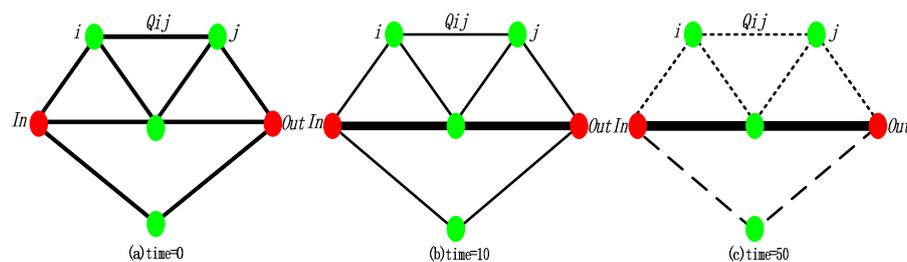


Figure 2. Schematic diagram of “pipeline culture” model of SMA.

- (1) Variables initialize.
- (2) Calculate the distance between each path point by distance formula. The distance between each path point is calculated by:

$$L_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{1/2} \tag{1}$$

where x_i, x_j represent the abscissa of i and j , and y_i, y_j represent the ordinate of i and j . L_{ij} is defined as the distance between i and j , also called the pipeline length of i to j .

- (3) Each path point can be regarded as a node in the pipeline network, and we will select two path points as the entrance point and the exit point, respectively. Then, the pressure value of each path point is calculated according to Kirchhoff’s Laws. The formula is expressed as:

$$\sum_i \frac{D_{ij}}{L_{ij}} (P_i - P_j) = \begin{cases} -I_0, & \text{for } j = 1 \\ I_0, & \text{for } j = 2 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where D_{ij} represents the conductivity of pipeline between i and j . P_i and P_j are the pressure of point i and j . The solution of conductivity needs to set a point pressure as the reference, and then calculate the rest of pressure of each point. For example, setting $p_2 = 0$ as the reference point of pressure.

(4) Q_{ij} is defined as the flow value between i and j . Q_{ij} needs to combine the relationship among the difference of pressure $(P_i - P_j)$, conductivity D_{ij} , and distance L_{ij} . The relation's formula that calculates the flow of each path pipeline is:

$$Q_{ij} = \frac{D_{ij}}{L_{ij}}(P_i - P_j) \tag{3}$$

(5) D_{ij} is the conductivity between i and j . The pipeline conductivity is required to be updated constantly. The calculation's formula is as follows:

$$\frac{dD_{ij}}{dt} = \frac{Q_{ij}}{1 + Q_{ij}} - D_{ij} \tag{4}$$

The iterative formula after deformation is:

$$D_{ij}(n + 1) = \left(\frac{|Q_{ij}(n)|}{1 + |Q_{ij}(n)|} - D_{ij}(n) \right) * \Delta t + D_{ij}(n) \tag{5}$$

(6) According to above process of (3) to (5), we complete the next cycle repeatedly until getting to the iteration termination condition. The stable value D_{ij} and Q_{ij} will be obtained by carrying out iteration, and the iterative condition of termination is defined as:

$$|D_{ij}(n + 1) - D_{ij}(n)| \leq \delta \tag{6}$$

(7) According to the final flow values, the next selected point is determined, and the path will be obtained by selecting the point of largest flow, from the starting point, one by one. After completing the selection of one point, the selected point P_{next} will become the current point i in the next point selection process. The point i is recorded into L_{best} by Equation (8), and the path result L_{best} is finally obtained by SMA. The concrete formula is shown as:

$$Q_{inext} = \max\{|Q_{i1}|, |Q_{i2}|, \dots, |Q_{im}|\} \tag{7}$$

$$L_{best} = \{i_1, i_2, \dots, i_n\} \tag{8}$$

where Q_{inext} represents the pipeline flow with the largest value from the current point i to the other points. At the same time, it is the flow of next path point selected. L_{best} is the path result by SMA.

3. Two-Way Parallel Slime Mold Algorithm by Flow and Distance

Due to the short development time, the research depth of SMA is insufficient and many details of the model have to be explored. As far as the "pipeline cultivation" model of SMA, the path point selection is only based on the flow value, and there is only one path to be obtained, so it makes the optimization results limited. Moreover, SMA cannot jump out of the local optimum, especially in the complex situation of points. Therefore, TPSMA is designed by using two-way parallel optimization on flow and distance in this paper. As shown in Figure 3, it uses two reference indicators of flow and distance to search for next point, instead of the original principle, which is selecting the next point only by flow. At the same time, a random factor is added when points are selected by the flow and distance, to increase the diversity of result.

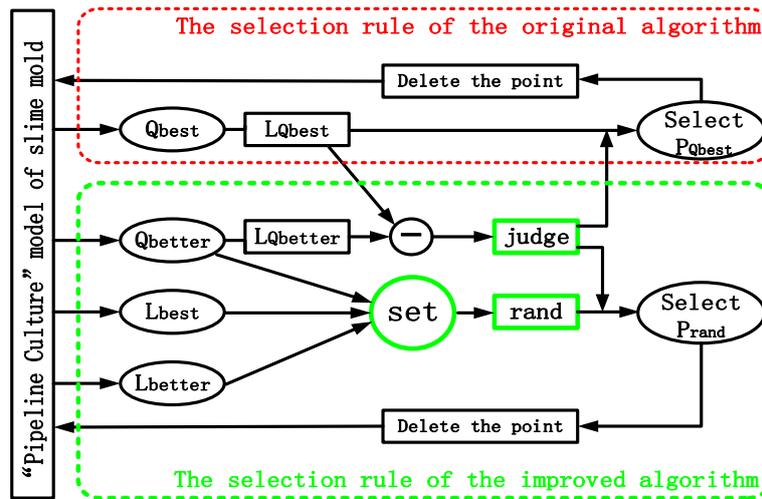


Figure 3. Schematic diagram of algorithm principle by TPSMA.

The schematic diagram of TPSMA is illustrated in Figure 3. Firstly, the flow value between points is obtained by using the “pipeline culture” model of SMA. Secondly, the path point can be selected by the flow and distance, according to the newly designed rule. The distance difference needs to be calculated between the path with the largest flow and the path with the second largest flow. If the distance difference is large, and the distance with the largest flow is shorter, the point with the largest flow will be selected as the next point. Otherwise, we need to form a point set that includes the shortest path point, the shorter path point, and the path point with the larger flow. Then, a random factor is added to achieve the selection of the next point when we select the points in point set. Finally, the optimized result of TPSMA will be obtained by iterating.

According to the principle of TPSMA on flow and distance, the algorithm is applied to the solution of TSP. As illustrated in Figure 4, the specific steps are as follows:

- (1) Variables initialize.
- (2) According to the Formulae (1)–(3), the distance L_{ij} and the flow Q_{ij} are calculated.
- (3) According to (2), we are going to complete the updating of conductivity and flow by Formula (5) and (3). Then, the final flow values are obtained by iterating until the condition (6) of termination is satisfied.
- (4) According to the two indexes of flow and distance, the point is selected in a two-way parallel method, and the specific contents are as follows:

(i) We select the Q_{ibest} of the largest flow value and the $Q_{ibetter}$ of the second largest flow value from the points to be selected and define the point of $P_{Q_{ibest}}$ and $P_{Q_{ibetter}}$. At the same time, the shortest distance L_{ibest} and the second shortest distance $L_{ibetter}$ are respectively found out and defined the point of $P_{L_{ibest}}$ and $P_{L_{ibetter}}$. As follows, $P_{Q_{ibetter}}$, $P_{L_{ibest}}$ and $P_{L_{ibetter}}$ are taken and formed a set P_{QL} . $P_{Q_{ibest}}$ is defined as the other choice of next point. The formula of point set is:

$$P_{QL} = \{P_{Q_{ibetter}}, P_{L_{ibest}}, P_{L_{ibetter}}\} \tag{9}$$

And the optional point is:

$$P_{Q_{ibest}} = \{P_{Q_{ibest}}\} \tag{10}$$

(ii) e_{LQ} is defined as the difference between $L_{Q_{ibest}}$ and $L_{Q_{ibetter}}$. If the distance of the path with the largest flow is too long or similar to others, the point in P_{QL} will be selected as the next point by random factors. Otherwise, $P_{Q_{ibest}}$, which is obtained by the maximum flow, is going to be chosen as

the next point. The improved rule, which is a two-way parallel selective method of flow and distance, is formulated by P_{next} , as expressed in (11) and (12):

$$e_{LQ} = L_{Qibest} - L_{Qibetter} \tag{11}$$

$$P_{next} = \begin{cases} Rand(P_{QL}) & , e_{LQ} > \varepsilon \\ P_{Qibest} & , e_{LQ} \leq \varepsilon \end{cases} \tag{12}$$

where L_{Qibest} is the distance of the path with the maximum flow, $L_{Qibetter}$ is the distance of the path with the second largest flow, P_{next} represents the next path point to be selected, $Rand(P_{QL})$ represents a random value in P_{QL} , which is the set of selected points, and ε is the determining parameter of distance difference. By adjusting the value of ε , which is usually a negative, an appropriate cut-off point will be obtained.

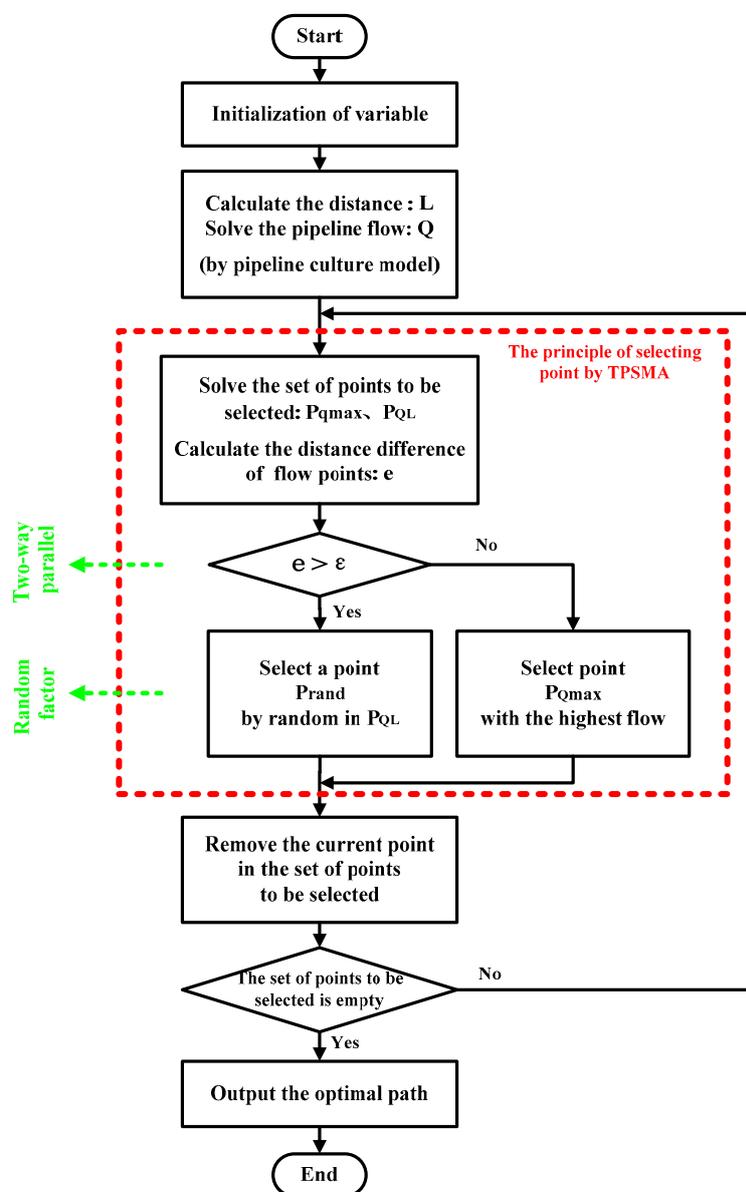


Figure 4. Flow chart of TPSMA.

(5) According to (4), a circulation will be completed when all the points are selected and the optimization result has been obtained. Then, the iterations are carried out until getting the termination condition. L_{length} , which is the set including all the paths, is obtained. Finally, the formula that can get the optimal path L_{best} in L_{length} is obtained by:

$$L_{best} = \min\{L_{length}\} \tag{13}$$

According to the above-mentioned steps, TPSMA is summarized as Table 1:

Table 1. The steps of TPSMA.

A Two-Way Parallel Slime Mold Algorithm by Distance and Flow	
input:	TSP path points
output:	TSP shortest path
(a)	Initialization process
Step 1	Initialize variables and parameters
(b)	Calculate the distance and flow
Step 2	Get the distance L_{ij} according to Formula (1)
Step 3	Get the flow Q_{ij} by the Formulas (2) and (3) of Kirchhoff's Laws
Step 4	Update the conductivity by Formula (5)
Step 5	Return to Step 3 to cycle until iterative condition is terminated and obtain the stable flow value Q_{ij}
(c)	The point selection by a two-way parallel method
Step 6	Select the points P_{Libest} , $P_{Libetter}$, P_{Qibest} and $L_{Qibetter}$, and gain two sets of points according to the Formula (9) and (10)
Step 7	Obtain the value e_{LQ} by subtracting L_{Qibest} from $L_{Qibetter}$. Then, complete the selection of point in turn by using the two-way parallel Formula (11) and (12)
Step 8	Return to Step 6 to finish the iterations until the iterative condition is satisfied, then get all the paths L_{length}
(d)	Obtain the result
Step 9	Obtain and output the optimal path L_{best} by the Formula (13)

4. Simulation and Analysis of Results

Traveling Salesman Problem Library (TSPLIB) is a library of sample instances for the TSP from various sources. Each set is a two-dimensional array containing horizontal and vertical coordinates of some cities, which is used as a test of intelligent algorithms. By using TSPLIB data to test and simulate in MATLAB, the experimental results were compared and analyzed to verify the effectiveness of TPSMA.

4.1. Result of Simulation

We selected four datasets of TSP in TSPLIB; ulysses16, city31, eil51, gr96, and bier127. Of these, city31 is the data set of longitude and latitude coordinates of the locations of 31 cities in China. Multiple sets of data are selected to ensure the reliability of the conclusions. At the same time, the number of the four groups of datasets is 16, 31, 51, 96, and 127, separately and increasing in order. It increases the difficulty of optimization and fully verifies the performance of the algorithm. Figures 5–9 are the simulation results obtained about five groups of TSP data by SMA and TPSMA. It can be seen from the figures that the results under the five groups of data are obviously improved by TPSMA.

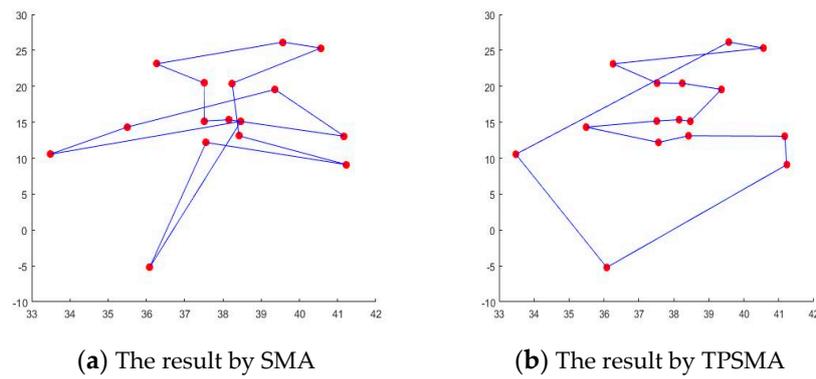


Figure 5. The simulation of ulysse16.

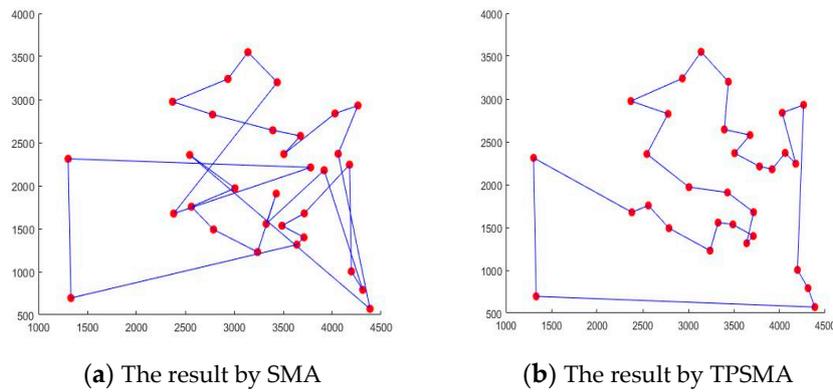


Figure 6. The simulation of city31.

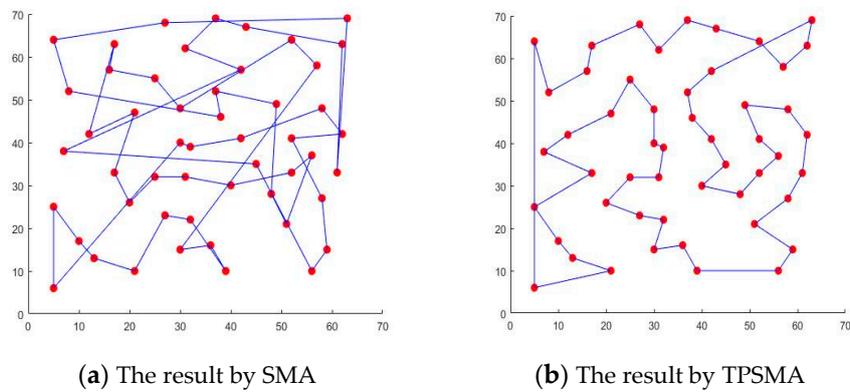


Figure 7. The simulation of eil51.

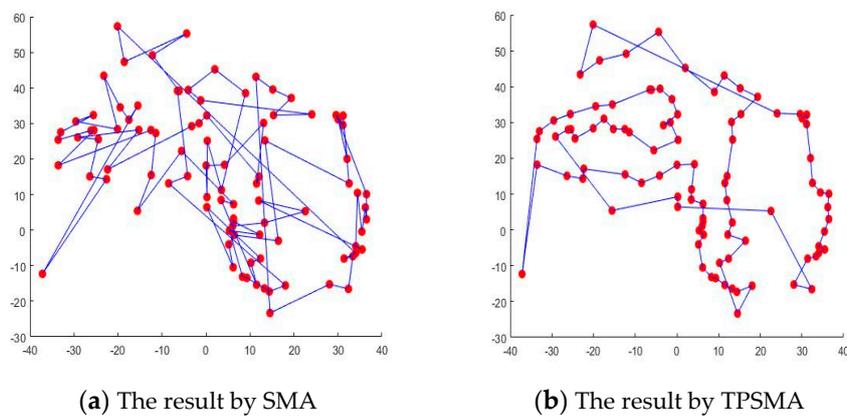


Figure 8. The simulation of gr96.

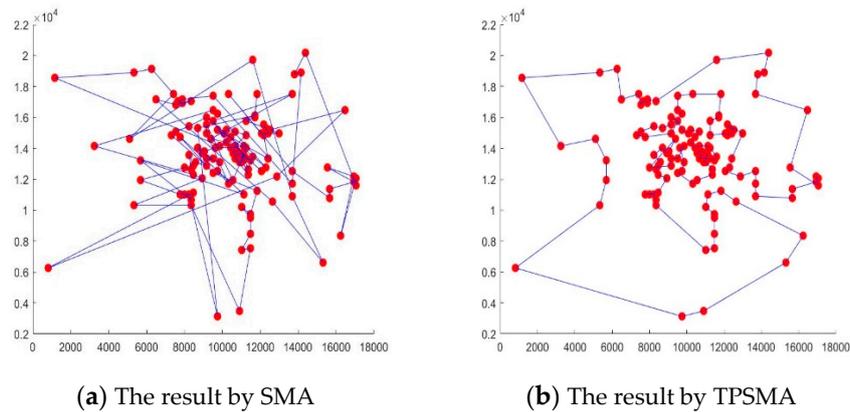


Figure 9. The simulation of bier127.

4.2. Analysis of Selection Processing

Taking ulysses16 data as an example, the optimization ability of TPSMA is verified by analyzing the optimization process. Figure 5 and Table 2 are simulations and partially obtained paths under ulysses16 data. In the Tables, red represents the current point and optimal result, green represents the selected point by SMA and TPSMA, and blue represents the remaining points in the set of candidate points by TPSMA. Figure 10 shows the flow and distance values from path point 12 to the remaining path points, and their ranking of each point.

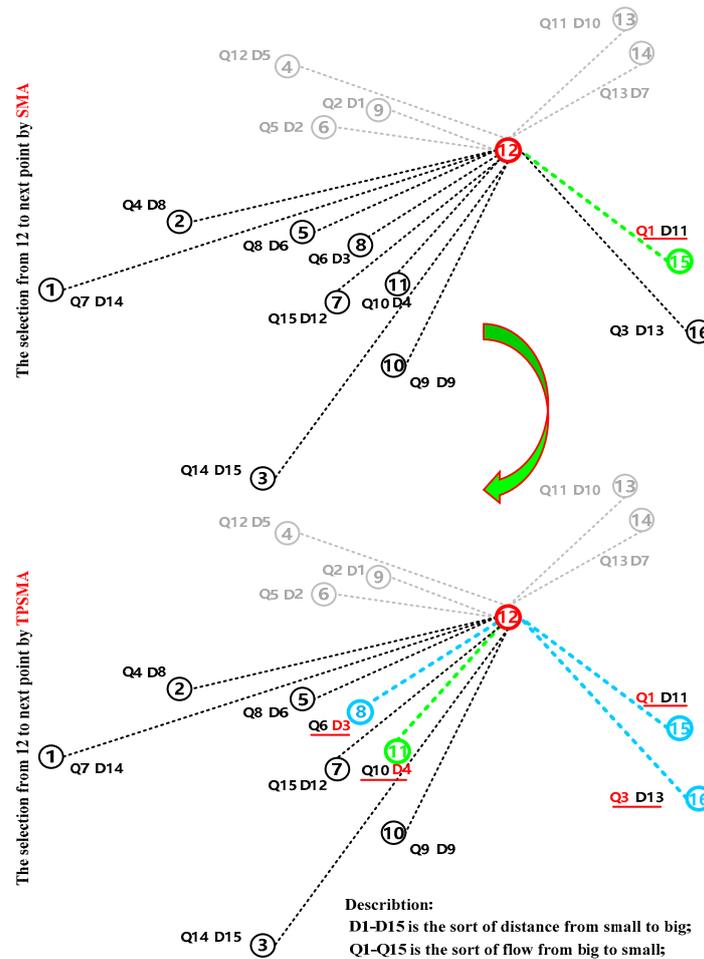


Figure 10. Data of flow and distance from point 12 to the remaining points, and the selection from 12 to the next point by SMA and TPSMA.

Table 2. Partial path optimization results under ulysses16 data.

No.	Path Optimization Results (Points Order)														Path Length		
1	13	14	4	6	9	12	11	8	5	2	7	10	15	16	3	1	77.8372
2	13	14	4	9	6	12	8	11	5	2	10	15	7	1	16	3	88.8287
3	13	14	4	9	6	12	15	16	1	2	7	10	11	8	5	3	94.3883
4	13	14	4	6	9	12	16	15	7	10	11	8	5	2	3	1	88.2858
5	14	9	10	6	7	3	11	1	2	12	15	8	5	6	4	13	103.174

When the current point is 12, it can be found in Figure 10 that the points of the largest flow, the second largest flow, the shortest distance, and the second shortest distance in the remaining candidate points are 15, 16, 8, and 11, respectively. In Table 2, point 15 with the maximum flow is selected by SMA, and the path 5 is the optimization result of SMA with a path length of 103.174. There are four candidate points, 15, 16, 8, and 11, which are obtained by TPSMA. Moreover, the path 1 to path 4 in Table 2 can be realized by adding random factors, and path 1 is the optimal result of TPAMA with a path length of 77.8372.

When the current point is point 3, it can be obtained in Figure 11 that the candidate points are 1, 2, and 7. Path 1 to path 5 in Table 3 can be obtained by random factors, and path 1 is the final result of TPSMA in this paper. Point 1 is selected after point 3 and the path length is 77.8372 in path 1. Furthermore, Table 3 shows that the next point of point 3 in path 2 is point 1 and the next choice of point 12 is point 11. The choices of two points is the same as the result of TPSMA in path 2. In contrast, only the next point of point 3 is the same as path 1 in path 3 and the next point of point 3 and point 12 in path 4 and path 5 is different from path 1. What is more, the results show that the path lengths of path 1, path 2, and path 3 are better than path 4 and path 5. From Table 3, it can be analyzed that the more points that selected by distance and flow, the better the result will be obtained. The above results show that TPSMA of two parameters with flow and distance is superior to SMA.

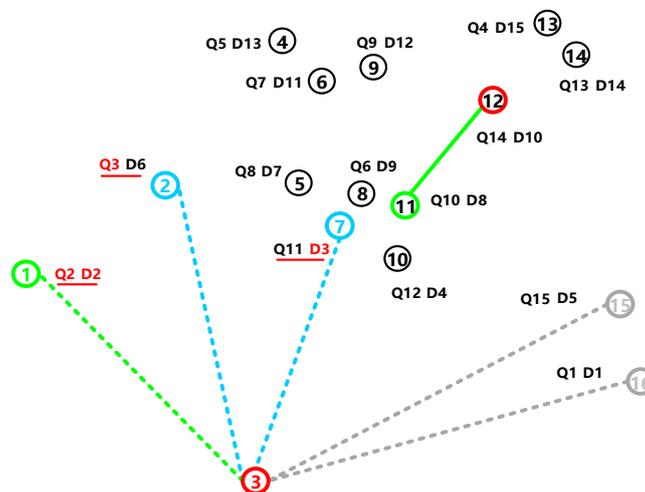


Figure 11. Data of flow and distance from point 3 to the remaining points.

Table 3. Partial path optimization results under ulysses16 data.

No.	Path Optimization Results (Points Order)														Path Length		
1	16	3	1	13	14	4	6	9	12	11	8	5	2	7	10	15	77.8372
2	16	3	1	2	14	13	4	9	6	12	11	8	15	10	5	7	80.9361
3	16	3	1	2	4	1	6	9	14	12	15	5	8	11	10	7	86.9835
4	16	3	2	1	4	13	6	9	14	12	15	11	5	7	8	10	98.9656
5	16	3	7	1	11	5	8	10	2	12	15	4	9	6	14	13	103.619

4.3. Analysis of Diversity

Under the data of ulysses16, as shown in Figure 12, SMA only gets path 1. However, the optimization process of TPSMA obtains the path 1 and path 2, which is the same as the length result of SMA. At the same time, in Figure 12, TPSMA gets two groups of paths 5 to 6 and paths 7 to 9. The path lengths of each group are the same but the paths have different orders of points. Similarly, TPSMA will produce a variety of cases due to the addition of random factors. Therefore, the diversity and comprehensiveness of the optimization results are increased.

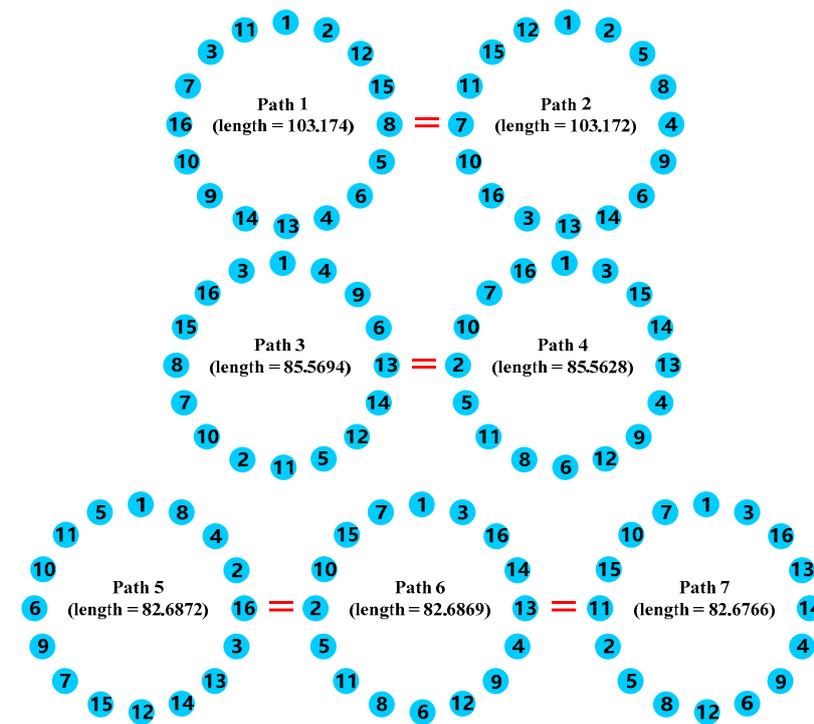


Figure 12. Partial paths of TPSMA under ulysses16 data.

4.4. Analysis of Optimization Ability in Different Numbers of Points

Based on four sets of data from ulysses16, city31, eil51, gr96, and bier127, the simulation diagrams are shown in Figures 5–9, and the results of data are shown in Table 4. Based on the analysis of the data, it can be obtained that:

Table 4. Experimental data’s results of ulysses16, city31, eil51, gr96 and bier127.

TSP Data	Results of SMA	Results of TPSMA	Improved Percentages
ulysses16	103.1746	77.8372	24.56%
city31	27,073	17,300	36.10%
eil51	798.9	464.3	41.88%
gr96	1178	591	49.83%
bier127	274,870	129,390	52.93%

Firstly, under the four sets of data, compared with SMA, the optimization ability of TPSMA is improved by 24.56, 36.10, 41.88, 49.83, and 52.93%, respectively. Therefore, the result of TPSMA is obviously better than SMA in Figure 13a.

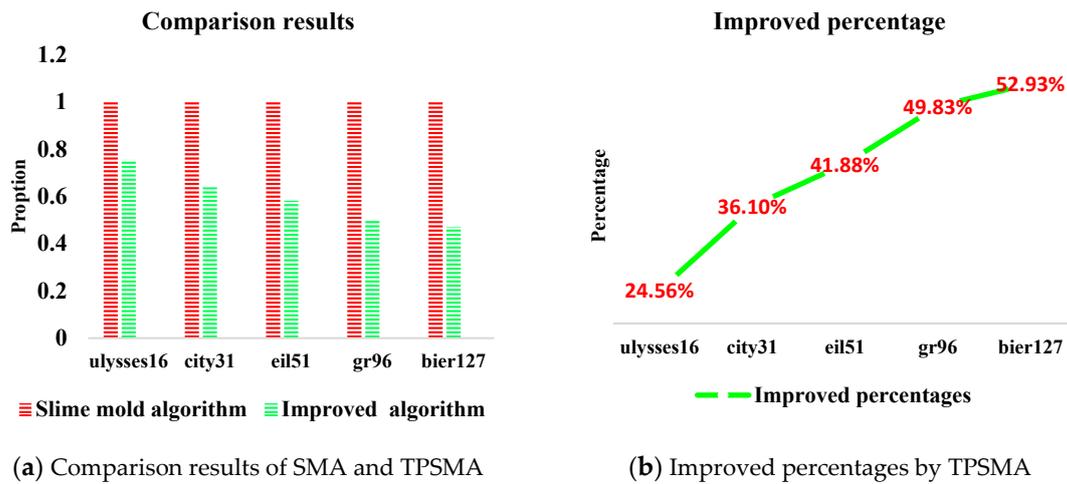


Figure 13. Analysis results of SMA and TPSMA under the four sets of data.

Secondly, as per the experimental results also shown in Figure 13b, the improved percentages of TPSMA are gradually increased with the increasing number of points. It can be seen that TPSMA designed in this paper is more powerful in solving TSP with large points and complex distribution. Thus, the experimental results fully verify the rationality and superiority of TPSMA.

4.5. Comparison with Other Algorithms

Based on some sets that have large points in TSPLIB, we can obtain the following by experiment. Table 5 is the results of each heuristic and bionic algorithm under different data sets. From the experimental results in Figure 14, the TPSMA results are obviously better than GA and PSO, and similar to ACO. Since the initial pheromone distribution of ACO is unpredictable, the reasonable distribution of pheromones needs to be gradually formed by iteration. Therefore, the convergence speed is very slow. Although TPSMA optimization results were slightly worse than ACO, TPSMA optimization speed was significantly better than ACO. Compared with the current best optimization result, which is from LKH, the result of TPSMA is not as good as LKH; however, it is closer to the optimal result than other algorithms. Moreover, LKH is in virtue of the 5 – opt principle which is based on the λ – opt algorithm. The more task points there are, the more iteration time it will cost, and the convergence performance will be poor.

Table 5. Comparisons of optimization results and algorithm features.

Name	PSO	GA	ACO	LKH	TPSMA	
Results of path length	eil51	1257	519	453	426	464
	eil76	2040	727	583	538	620
	lin105	96,429	30,167	15,303	14,379	16,424
	bier127	542,558	196,276	128,147	118,282	129,390
	kroA200	290,368	87,786	33,471	29,368	34,972
	gil262	23,780	7769	2779	2378	2881
Algorithm Time Complexity	N^3	N^3	N^4	N^5	N^3	
Theoretical difficulty	Complexity	Simple	Complexity	Complexity	Simple	

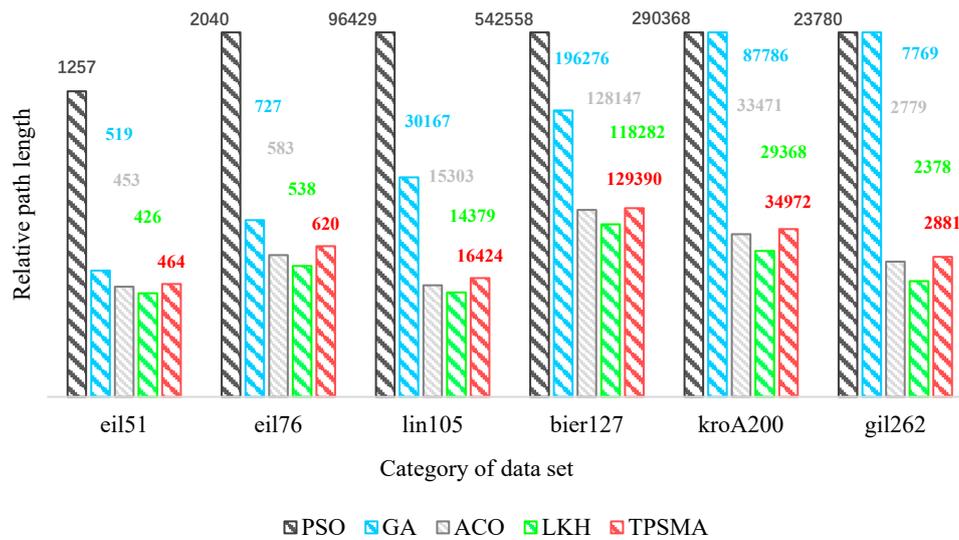


Figure 14. Comparison of optimization results under TSPLIB data sets.

We can get the degree of difficulty about algorithm principle and the algorithm time complexity in Table 5. It can be seen that the TPSMA algorithm is simple in principle and easy to analyze. At the same time, the algorithm time complexity of TPSMA (N^3) is less $1/N$ times than that of ACO (N^4), and less $1/N^2$ times than that of LKH (N^5). TPSMA has fast convergence speed and short optimization time due to low algorithm time complexity, especially in large data points.

According to the above analysis, it can be concluded from Figure 15 that TPSMA has good searching ability by a unique way, and fast optimization speed by low time complexity. At the same time, TPSMA is simple in theory and easy to research, therefore it is better to study than the others. Furthermore, the research time of TPSMA is short and the algorithm is not mature enough—there are many performances and potential to be developed due to its unique optimization method and effectiveness.

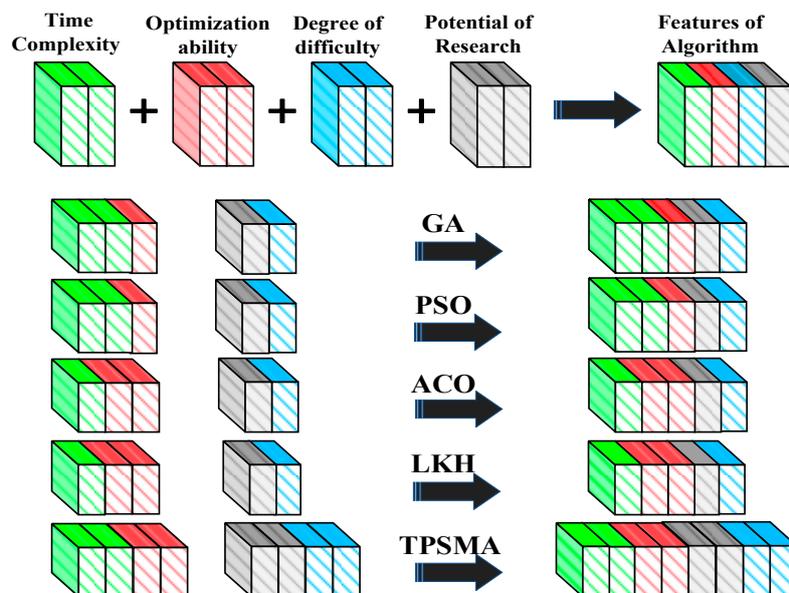


Figure 15. Comparison and analysis of algorithm characteristics.

5. Summary

In this paper, the two-way parallel selection principle of distance and flow is adopted by TPSMA, and random factor is added to improve optimization ability and diversity. Through the experimental results of TSPLIB data, the path length that is obtained by TPSMA is obviously reduced. What is more, the optimization ability is gradually enhanced with the increase in the number of path points. At the same time, TPSMA can get all the paths that meet the requirements to realize the diversity of path results. The above results prove the feasibility and superiority of TPSMA in solving TSP.

The proposed method will show partial reversal paths and diagonal paths in the searching process, which could have an impact on the search results. Follow-up research can start with the direction of flow, to research and improve the algorithm performance.

Author Contributions: Conceptualization, M.L., Y.L. and Q.H.; methodology, M.L. and Y.L.; software A.L.; validation, Y.L.; formal analysis, M.L. and Y.L.; investigation, Q.H. and M.X.; resources, M.Z. and L.C.; data curation, Q.H.; writing—original draft preparation, M.L.; writing—review and editing, M.L., M.Z. and N.Q.; visualization, M.X.; supervision, L.C.; project administration, N.Q.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 11672290), Science and Technology Development Plan of Jilin province (No. 2018020102GX), and Jilin Province and the Chinese Academy of Sciences cooperation in science and technology high-tech industrialization special funds project (No. 2018SYHZ0004).

Conflicts of Interest: The authors declare no conflict of interest. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

- Lu, Y.; Liu, Y.; Gao, C.; Tao, L.; Zhang, Z. A Novel *Physarum*-Based Ant Colony System for Solving the Real-World Traveling Salesman Problem. In Proceedings of the International Conference in Swarm Intelligence, Hefei, China, 17–20 October 2014; Volume 8794, pp. 173–180.
- Ouaarab, A.; Ahiod, B.; Yang, X.S. Random-key cuckoo search for the travelling salesman problem. *Soft Comput.* **2015**, *19*, 1099–1106. [[CrossRef](#)]
- Ahmadi, E.; Goldengorin, B.; Gürsel, S.; Mosadegh, H. A hybrid method of 2-TSP and novel learning-based GA for job sequencing and tool switching problem. *Appl. Soft Comput.* **2018**, *65*, 214–229. [[CrossRef](#)]
- Khan, I.; Pal, S.; Maiti, M.K. A Hybrid PSO-GA Algorithm for Traveling Salesman Problems in Different Environments. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2019**, *27*, 93–117. [[CrossRef](#)]
- Sengupta, S.; Basak, S.; Peters, R. Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. *Mach. Learn.* **2018**, *1*, 157–191. [[CrossRef](#)]
- Zhang, C.; Li, Q.; Chen, P.; Feng, Q.; Cui, J. An improved multi-objective particle swarm optimization and its application in raw ore dispatching. *Adv. Mech. Eng.* **2018**, *10*, 168781401875737. [[CrossRef](#)]
- Yu, M. A solution of TSP based on the ant colony algorithm improved by particle swarm optimization. *Discret. Contin. Dyn. Syst.* **2019**, *12*, 979–987. [[CrossRef](#)]
- Nechita, E.; Crisan, G.C.; Lantovics, L.B.; Huang, Y. On the Resilience of Ant Algorithms. Experiment with Adapted MMAS on TSP. *Mathematics* **2020**, *8*, 752. [[CrossRef](#)]
- Karapetyan, D.; Gutin, G. Lin-Kernighan Heuristic Adaptations for the Generalized Traveling Salesman Problem. *Eur. J. Oper. Res.* **2011**, *208*, 221–232. [[CrossRef](#)]
- Kernighan local search for a real-life Two-Echelon Heterogeneous Vehicle Routing Problem based on Brazilian wholesale companies. *Appl. Soft Comput.* **2019**, *76*, 697–711. [[CrossRef](#)]
- Helsgaun, K. General k-opt submoves for the Lin-Kernighan TSP heuristic. *Math. Program. Comput.* **2009**, *1*, 119–163. [[CrossRef](#)]
- Helsgaun, K. Solving Arc Routing Problems Using the Lin-Kernighan-Helsgaun Algorithm. *ICE Trans. Inf. Syst.* **2015**, *95*, 1031–1043.
- Stephenson, S.L.; Stempen, H. Myxomycetes: A Handbook of Slime Molds by Steven, L. Stephenson. *New Phytol.* **1996**, *132*, 201.
- Yamada, H.; Tóth, G.; Nakagaki, T. Intelligence: Maze-Solving by an Amoeboid Organism. *Nature* **2000**, *407*, 470.

15. Tero, A.; Takagi, S.; Saigusa, T.; Ito, K.; Bebbler, D.P.; Fricker, M.D.; Yumiki, K.; Kobayashi, R.; Nakagaki, T. Rules for Biologically Inspired Adaptive Network Design. *Science* **2010**, *327*, 439–442. [[CrossRef](#)] [[PubMed](#)]
16. Strano, E.; Adamatzky, A.; Jones, J. Physarum Itineraria: Evolution of Roman Roads with Slime Mould. *Int. J. Nanotechnol. Mol. Comput.* **2011**, *3*, 31–55. [[CrossRef](#)]
17. Adamatzky, A.; Yang, X.S.; Zhao, Y.X. Slime mould imitates transport networks in China. *Int. J. Intell. Comput. Cybern.* **2013**, *6*, 232–251. [[CrossRef](#)]
18. Evangelidis, V.; Tsompanas, M.; Sirakoulis, G.C.; Adamatzky, A. Slime mould imitates development of Roman roads in the Balkans. *J. Archaeol. Sci.* **2015**, *2*, 264–281. [[CrossRef](#)]
19. Becker, M. On the quality of graphs generated by swarm algorithms. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016; pp. 3734–3739.
20. Meyer-Nieberg, S. Coordinating a team of searchers: Of ants, swarms, and slime molds. In Proceedings of the Genetic & Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 27–34.
21. Zhang, X.; Mahadevan, S. A Bio-Inspired Approach to Traffic Network Equilibrium Assignment Problem. *IEEE Trans. Cybern.* **2018**, *48*, 1304–1315. [[CrossRef](#)]
22. Takaoka, T.; Sato, M.; Otake, T. Novel Routing Method Using Slime Mold Algorithm Corresponding to Movement of Content Source in Content-Oriented Networks. *J. Signal Process.* **2019**, *23*, 173–176. [[CrossRef](#)]
23. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
24. Nakagaki, T.; Yamada, H.; Hara, M. Smart network solutions in an amoeboid organism. *Biophys. Chem.* **2004**, *107*, 1–5. [[CrossRef](#)]
25. Tero, A.; Kobayashi, R.; Nakagaki, T. Physarum solver: A biologically inspired method of road-network navigation. *Phys. A Stat. Mech. Appl.* **2006**, *363*, 115–119. [[CrossRef](#)]
26. Kobayashi, R.; Tero, A.; Nakagaki, T. Mathematical Model for Rhythmic Protoplasmic Movement in the True Slime Mold. *J. Math. Biol.* **2006**, *53*, 273–286. [[CrossRef](#)] [[PubMed](#)]
27. Tero, A.; Kobayashi, R.; Nakagaki, T. A mathematical model for adaptive transport network in path finding by true slime mold. *J. Theor. Biol.* **2007**, *244*, 553–564. [[CrossRef](#)]
28. Nakagaki, T.; Iima, M.; Ueda, T. Minimum-risk path finding by an adaptive amoebal network. *Phys. Rev. Lett.* **2007**, *99*, 068104. [[CrossRef](#)] [[PubMed](#)]
29. Gunji, Y.P.; Shirakawa, T.; Niizato, T. Minimal model of a cell connecting amoebic motion and adaptive transport networks. *J. Theor. Biol.* **2008**, *253*, 659–667. [[CrossRef](#)] [[PubMed](#)]
30. Gunji, Y.P.; Shirakawa, T.; Niizato, T. An adaptive and robust biological network based on the vacant-particle transportation model. *J. Theor. Biol.* **2011**, *272*, 187–200. [[CrossRef](#)]
31. Liu, Y.; Lu, Y.; Gao, C.; Zhang, Z.; Tao, L. A multi-objective ant colony optimization algorithm based on the Physarum-inspired mathematical model. In Proceedings of the 10th International Conference on Natural Computation, Xiamen, China, 19–21 August 2014; pp. 303–308.
32. Zhang, Z.; Gao, C.; Liu, Y.; Qian, T. A universal optimization strategy for ant colony optimization algorithms based on the Physarum-inspired mathematical model. *Bioinspiration Biomim.* **2014**, *9*, 036006. [[CrossRef](#)]
33. Zhang, Z.; Gao, C.; Lu, Y.; Liu, Y.; Liang, M. Multi-Objective Ant Colony Optimization Based on the Physarum-Inspired Mathematical Model for Bi-Objective Traveling Salesman Problems. *PLoS ONE* **2016**, *11*, e0146709. [[CrossRef](#)]
34. Liu, Y.; Gao, C.; Zhang, Z. A new multi-agent system to simulate the foraging behaviors of Physarum. *Nat. Comput.* **2017**, *16*, 15–19. [[CrossRef](#)]
35. Adamatzky, A. Route 20, Autobahn 7, and Slime Mold: Approximating the Longest Roads in USA and Germany with Slime Mold on 3-D Terrains. *IEEE Trans. Cybern.* **2014**, *44*, 126–136. [[CrossRef](#)] [[PubMed](#)]
36. Uemura, M.; Matsushita, H.; Kraetzschmar, G.K. Path Planning with Slime Molds: A Biology-Inspired Approach. In Proceedings of the International Conference on Neural Information Processing, Istanbul, Turkey, 9–12 November 2015; Volume 9492, pp. 308–315.
37. Adamatzky, A. Physical maze solvers. All twelve prototypes implement 1961 Lee algorithm. *Emergent Comput.* **2016**, *24*, 489–504.

